University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: July 21, 2024

Group Number: 26

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Andrew Munro-West	18363572	i6z2b (amunwes)	andrewmunrowest@gmail.com
Andrew Ehebald	45888054	a2l2b (aehebald)	andrewehebald@outlook.com
Laura Qiu	10810190	k3v1u (lqiu05)	lauraqiu21@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

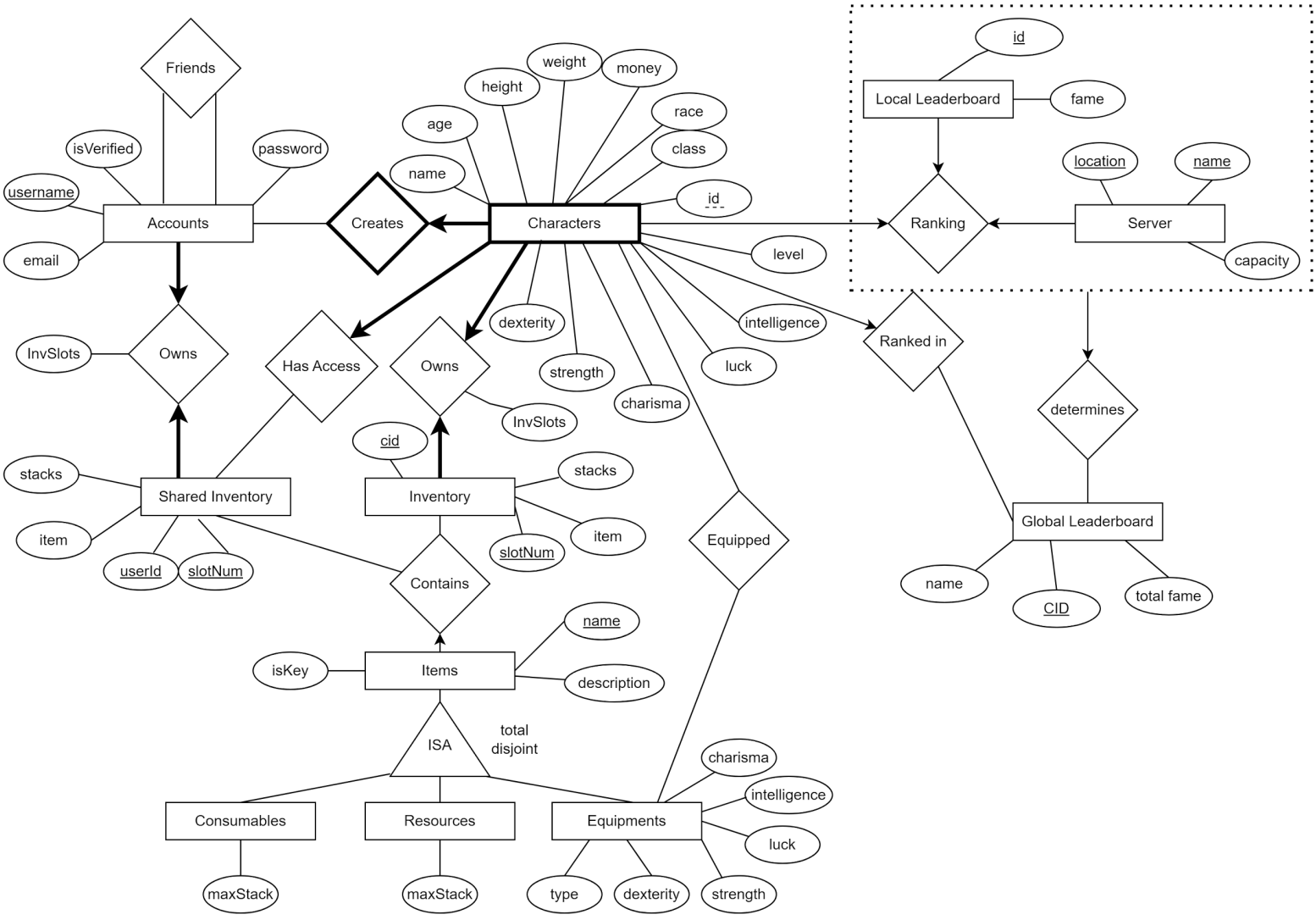
University of British Columbia, Vancouver
Department of Computer Science

2) Description

For this project we are proposing a user database for an MMORPG (massive multiplayer online role playing game) similar in scale and scope to well known titles like “the world of warcraft” and “final fantasy online”. In theory this would mean that the database would be responsible for the player data of 10,000,000+ player accounts as well as the inventory status and ranking of any characters created under a player account. As such the project would fall under the domains of Entertainment/Gaming, User/Account management, Inventory tracking, and leaderboard tracking.

University of British Columbia, Vancouver
Department of Computer Science

3)ER diagram



We took our mentor's advice and removed the appearance entity, moving its attributes into characters and instead made characters a weak entity.
We renamed a few attributes, and moved the inv slots attribute into the owns relationship instead of being in the inventory entity.
Money was moved out of inventory and put on character.

University of British Columbia, Vancouver
Department of Computer Science

4)The Schema

Table Name	Attributes	Constraints
Accounts	username:VARCHAR(20), isVerified:BOOL(), password:VARCHAR(20), email:VARCHAR(320), InvSlots:INT(3)	username is PK, email is UNIQUE and NOT NULL, password, invSlots is NOT NULL,
Friends	user:VARCHAR(20), friend:VARCHAR(20)	user is PK, user and friend are FK referencing accounts username cascade on delete,
Characters	ID:CHAR(20), AccName: VARCHAR(20), name:VARCHAR(20), class:VARCHAR(20), age:INT(2), height:INT(3), weight: INT(3), race:VARCHAR(20), level: int(5), money: INT(15), strength: INT(5), intelligence:INT(5), charisma:INT(5), dexterity:INT(5), luck:INT(5),InvSlots:INT(2)	ID and AccName are PK AccName is FK to Account's Username On Delete Cascade, name, class, age, height, weight,race, level, money, strength, intelligence, charisma, dexterity, luck, InvSlots are NOT NULL strength, intelligence, charisma, dexterity, luck, level DEFAULT is 1 money DEFAULT is 0
Equipped	EqName:VARCHAR(20), CID:CHAR(20), AccName: VARCHAR(20), EqType:VARCHAR(20)	EqName, CID, AccName are PK CID,AccName is FK to character's ID,AccName EqName,EqType are FK referencing equipments Name,Type Cascade on delete of CID (CID,EqType) is CK and UNIQUE
Inventory	CID:CHAR(20), AccName: VARCHAR(20), slotNum:INT(3), stacks:INT(4), item: VARCHAR(20)	CID,AccName, slotNum is PK CID, AccName is FK to character's ID,AccName Cascade ON Delete item is FK to item's name stacks DEFAULT = 1

University of British Columbia, Vancouver
Department of Computer Science

SharedInventory	userID:VARCHAR(20),slotNum:INT(3), stacks:INT(4), item: VARCHAR(20)	userID, slotNum is PK userID is FK referencing Account's Username, ON Delete Cascade, item is FK to item's name, stacks Default = 1
Items	name:VARCHAR(20), isKey:BOOL(), description: MEDIUMTEXT()	name is PK isKey DEFAULT is 0
Consumables	name:VARCHAR(20), maxStack:int(2)	name is PK and FK to items name Cascade on delete, Cascade on update
Equipments	name:VARCHAR(20), type:VARCHAR(20), strength: INT(5), intelligence:INT(5), charisma:INT(5), dexterity:INT(5), luck:INT(5),	name is PK and FK to items nameCascade on delete, Cascade on Update type is NOT NULL
Resources	name:VARCHAR(20), maxStack:int(4)	name is PK and FK to items name Cascade on delete, Cascade on update
GlobalLeaderboard	totalFame: INT(15), rank:INT(10), CID: CHAR(20), AccName: VARCHAR(20), name:VARCHAR(20)	CID, AccName is PK rank is UNIQUE, CID,AccName,name is FK to character's ID,AccName,name, on delete Cascade.
LocalLeaderBoard	rank:int(10), ID:CHAR(20), Fame: INT(15), CID:CHAR(20), AccName: VARCHAR(20), location:VARCHAT(20)	id, rank is PK CID,AccName is FK to character's ID,AccName Cascade on delete. location is FK to server's location Cascade on update, Cascade on delete
Server	name:VARCHAR(20), location:VARCHAR(20), capacity:INT(6)	name and location are PK

Additional constraints to query before insert not related to the table:
- an account can be unverified but a character cannot be created until an account is verified
- must check and decrement character/Account Inv slots before inserting into Inventory/sharedInventory or increment when deleting rows.
- Global leaderboard's total fame is found by querying entries of LocalLeaderBoard for all entries of a single character and summing the fame column.

University of British Columbia, Vancouver
Department of Computer Science

5) Functional dependencies
FD's:
Accounts:
username -> isVerified, password, email, InvSlots
email->username, password, isVerified
friends:
friend-> user
user-> friend
Characters:
ID, AccName-> …
ID->money
race->height, weight
class, level -> strength,intelligence,dexterity,charisma,luck
Equipped:
EqName, CID, AccName -> EqType
EqName -> EqType
Inventory:
CID, AccName, SlotNum-> item
item->stacks
SharedInventory:
userID,SlotNum -> item
item-> stacks
Items:
name-> description, isKey
description -> name
Consumables:
name-> maxStacks
Equipments:
name -> maxStacks,Type,....
Resources:
name->maxStacks
GlobalLeaderboard:
CID, AccName -> totalFame, name, rank
rank->totalFame,name,CID
LocalLeaderboard:
id,rank->CID,fame,location
id->location
location, rank -> fame
CID,AccName, location -> fame
Server:
name,location -> capacity
name -> capacity

University of British Columbia, Vancouver
Department of Computer Science

6) Normalization
Characters:
ID, AccName-> name, age, height, weight, money, race, class, level, intelligence, luck, charisma, strength, dexterity,
ID->AccName, money
race->height, weight
class, level -> strength,intelligence,dexterity,charisma,luck

Find minimal cover:

1) Put FD's in standard form

ID, AccName -> name
ID, AccName -> age
ID, AccName -> height
ID, AccName -> weight
ID, AccName -> money
ID, AccName -> race
ID, AccName -> class
ID, AccName -> level

ID, AccName -> intelligence
ID, AccName -> luck
ID, AccName -> charisma
ID, AccName -> strength
ID, AccName -> dexterity

ID-> money

race -> height
race - > weight
class,level -> strength
class,level -> intelligence
class,level -> dexterity
class,level -> charisma
class,level -> luck

Step 2 Minimize LHS for each FD:
ID+ = { ID, money }
AccName+ = {AccName}

Since ID and AccName can not determine any additional attributes the following attribute are in minimal form:
ID, AccName -> name
ID, AccName -> age

ID, AccName -> height
ID, AccName -> weight
ID, AccName -> race
ID, AccName -> class
ID, AccName -> level
ID, AccName -> intelligence
ID, AccName -> luck
ID, AccName -> charisma
ID, AccName -> strength
ID, AccName -> dexterity

However, ID, AccName -> money can be simplified to ID -> money since the closure of ID includes money

ID -> money, race -> weight, race -> height : requires only a single attribute so it is in minimal form by default

class+ = {class}
level = {level}

Since these closures cannot determine any other attributes the following are in minimal form:

class,level -> strength
class,level -> intelligence
class,level -> dexterity
class,level -> charisma
class,level -> luck

Step 3: Delete redundant FDs

The FDs after minimizing FDs are:

ID, AccName -> name
ID, AccName -> age
ID, AccName -> height
ID, AccName -> weight
ID -> money
ID, AccName -> race
ID, AccName -> class
ID, AccName -> level
ID, AccName -> intelligence
ID, AccName -> luck
ID, AccName -> charisma

ID, AccName -> strength
ID, AccName -> dexterity

ID-> money

race -> height
race - > weight
class,level -> strength
class,level -> intelligence
class,level -> dexterity
class,level -> charisma
class,level -> luck

Now there are two ID -> money FDs so we can delete one of those.

We can see that we can safely remove
ID, AccName - > height
ID,AccName -> weight
ID,AccName - >strength
ID, AccName -> intelligence
ID,AccName -> luck
ID, AccName -> charisma
ID, AccName - > dexterity

So the minimal cover of the following would be:

ID, AccName -> name
ID, AccName -> age
ID, AccName -> race
ID, AccName -> class
ID, AccName -> level
ID -> money
race -> height
race - > weight
class,level -> strength
class,level -> intelligence
class,level -> dexterity
class,level -> charisma
class,level -> luck

Now we find the minimal keys:

University of British Columbia, Vancouver
Department of Computer Science

Left	Middle	Right
ID, AccName	race, class, level	name, age, money, strength, intelligence, dexterity, charisma, luck, height, weight

ID,AccName + = { ID, AccName, name, age, race, class, level, money, height, weight, strength, intelligence, dexterity, charisma, luck}

We can see that ID, AccName is a minimal key

We now check to see if our relation is already in 3NF:

We define 3NF as: X-> b is a non-trivial dependency => X is a super key or b is part of the key

We see that ID -> money is not in 3NF as this is not a trivial dependency and ID is not a super key and money is not part of the minimal key, so we decompose the relation using BCNF:

R (ID, AccName, name, age, race, class, level, money, height, weight, strength, intelligence, dexterity, charisma, luck)
We decompose into:

R1(ID, AccName, name, age, race, class, level,height, weight, strength, intelligence, dexterity, charisma, luck) and R2(ID, money)

We see that race -> height violates 3NF so we continue our decomposition as :
R2(ID, money)
R3(ID, AccName, name, age, race, class, level,weight, strength, intelligence, dexterity, charisma, luck)
R4(race, height)

We see that race -> weight violates 3NF so we continue our decomposition as:

R2(ID, money)
R4(race, height)
R5(ID, AccName, name, age, race, class, level, strength, intelligence, dexterity, charisma, luck)
R6 (race, weight)

We see that class,level -> strength violates 3NF so we continue our decomposition as:
R2(ID, money)
R4(race, height)

R6 (race, weight)
R7(ID, AccName, name, age, race, class, level,, intelligence, dexterity, charisma, luck)
R8(class, level, strength)

We see that class,level -> intelligence violated 3NF so we continue our decomposition as:
R2(ID, money)
R4(race, height)
R6 (race, weight)
R8(class, level, strength)
R9(ID, AccName, name, age, race, class, level, dexterity, charisma, luck)
R10(class, level, intelligence)

We see that class,level -> dexterity violated 3NF so we continue our decomposition as:
R2(ID, money)
R4(race, height)
R6 (race, weight)
R8(class, level, strength)
R10(class, level, intelligence)
R11(ID, AccName, name, age, race, class, level, dexterity, charisma, luck)
R12(class, level, dexterity)

We see that class,level -> charisma violated 3NF so we continue our decomposition as:
R2(ID, money)
R4(race, height)
R6 (race, weight)
R8(class, level, strength)
R10(class, level, intelligence)
R12(class, level, dexterity)
R13(ID, AccName, name, age, race, class, level, charisma, luck)
R14(class, level, charisma)

We see that class,level -> luck violated 3NF so we continue our decomposition as:
R2(ID, money)
R4(race, height)
R6 (race, weight)
R8(class, level, strength)
R10(class, level, intelligence)
R12(class, level, dexterity)
R14(class, level, charisma)
R15(**ID**, AccName, name, age, **race**, **class**, **level**)
R16(class, level, luck)

University of British Columbia, Vancouver
Department of Computer Science

LocalLeaderboard:
id,rank->CID,fame,location
id->location
id, rank->fame
location, rank -> fame
CID,AccName, location -> fame

Minimal Cover:

1) Put all relations in normal form:
   id, rank -> CID
   id, rank -> frame
   id,rank -> location
   id -> location
   location, rank -> fame
   CID, AccName, location -> fame
2) minimize LHS of relations:

   We see that
   id+ = { id, location}

   So we do not need rank in the id, rank -> location dependency.

   After that reduction we have:
   id, rank -> CID
   id, rank -> frame
   id -> location
   id -> location
   location, rank -> fame
   CID, AccName, location -> fame

3) We now remove redundant relations:
   We see that we have duplicate id -> location relations so we can delete one.

   We have two relations that both can determine fame so we check:

   Can we determine fame without location, rank -> fame?
   location, rank + = {location, rank }

   We cannot so this is not redundant

Can we determine fame without CID, Accname, location -> fame?

CID, Accname, location+ = {CID, Accname, location}

We cannot so this is not redundant

So the minimal cover is:

id, rank -> CID
id, rank -> fame
id -> location
location, rank -> fame
CID, AccName, location -> fame

Now that we have the minimal cover we not check if this is in 3NF:

Now we find the minimal keys:

Left	Middle	Right
id, rank, AccName	CID, location	fame

id, rank, AccName + = {id, rank, AccName, CID, fame, location}

So id, rank, AccName is the minimal key

We notice that id, rank -> CID violates 3NF so we decompose:

R1(id, rank, CID) R2(id, rank, fame) R3(id, location) R4(location, rank, fame) R5(CID, AccName, location, fame)

We now add a key dependency as well:
R6(id, rank, AccName)

So after normalization we get

R1(<u>id</u>, <u>rank</u>, CID) R2(<u>id</u>, <u>rank</u>, fame) R3(<u>id</u>, location) R4(<u>location</u>, <u>rank</u>, fame) R5(<u>CID</u>, <u>AccName</u>, <u>location</u>, fame) R6(**<u>id, rank, AccName</u>**)

7. The SQL DDL statements for tables

```
CREATE TABLE Accounts (
        username VARCHAR(20) PRIMARY KEY,
        isVerified BOOLEAN,
        password VARCHAR(20) NOT NULL,
        email VARCHAR(320) UNIQUE NOT NULL,
        InvSlots INT(3) NOT NULL
);

CREATE TABLE Friends {
        user VARCHAR(20) PRIMARY KEY,
        friend VARCHAR(20),
        FOREIGN KEY (user) REFERENCES Accounts(username) ON DELETE CASCADE,
        FOREIGN KEY (friend) REFERENCES Accounts(username) ON DELETE CASCADE
}

CREATE TABLE Characters (
        ID VARCHAR(20),
        AccName VARCHAR(20),
        name VARCHAR(20) NOT NULL,
        class VARCHAR(20) NOT NULL,
        age INT(2) NOT NULL,
        height INT(3) NOT NULL,
        weight INT(3) NOT NULL,
        race VARCHAR(20) NOT NULL,
        level INT(5) NOT NULL DEFAULT 1,
        money INT(15) NOT NULL DEFAULT 0,
        strength INT(5) NOT NULL DEFAULT 1,
        intelligence INT(5) NOT NULL DEFAULT 1,
        charisma INT(5) NOT NULL DEFAULT 1,
        dexterity INT(5) NOT NULL DEFAULT 1,
        luck INT(5) NOT NULL DEFAULT 1,
        InvSlots INT(2) NOT NULL,
        PRIMARY KEY (ID, AccName),
        FOREIGN KEY (AccName) REFERENCES Accounts(username) ON DELETE CASCADE
);

CREATE TABLE Equipped (
        EqName VARCHAR(20),
        CID VARCHAR(20),
        AccName VARCHAR(20),
        EqType VARCHAR(20),
        PRIMARY KEY (EqName, CID, AccName),
        FOREIGN KEY (CID, AccName) REFERENCES Characters(ID, AccName) ON DELETE CASCADE,
        FOREIGN KEY (EqName, EqType) REFERENCES Equipments(name, type) ON DELETE CASCADE,
        UNIQUE (CID, EqType)
);

CREATE TABLE Inventory {
        CID VARCHAR(20),
        AccName VARCHAR(20),
        slotNum INT(3),
```

```
        stacks INT(4) DEFAULT 1,
        item VARCHAR(20),
        PRIMARY KEY (CID, AccName, slotNum),
        FOREIGN KEY (CID, AccName) REFERENCES Character(ID, AccName) ON DELETE CASCADE,
        FOREIGN KEY(item) REFERENCES Items(name)
}

CREATE TABLE SharedInventory {
        userID VARCHAR(20),
        slotNum INT(3),
        stacks INT(4) DEFAULT 1,
        item VARCHAR(20),
        PRIMARY KEY(userID, slotNum),
        FOREIGN KEY(userID) REFERENCES Accounts(userName) ON DELETE CASCADE,
        FOREIGN KEY(item) REFERENCES Items(name)
}

CREATE TABLE Items {
        name VARCHAR(20) PRIMARY KEY,
        isKey BOOLEAN DEFAULT 0,
        description MEDIUMTEXT
}

CREATE TABLE Consumables {
        name VARCHAR(20) PRIMARY KEY,
        maxStack INT(4),
        FOREIGN KEY (name) REFERENCES Items(name) ON DELETE CASCADE ON UPDATE CASCADE
}

CREATE TABLE Equipments {
        name VARCHAR(20) PRIMARY KEY,
        type VARCHAR(20) NOT NULL,
        strength INT(5),
        intelligence INT(5),
        charisma INT(5),
        dexterity INT(5),
        luck INT(5),
        FOREIGN KEY (name) REFERENCES Items(name) ON DELETE CASCADE ON UPDATE CASCADE
}

CREATE TABLE Resources {
        name VARCHAR(20) PRIMARY KEY,
        maxStack INT(4),
        FOREIGN KEY (name) REFERENCES Items(name) ON DELETE CASCADE ON UPDATE CASCADE
}

CREATE TABLE GlobalLeaderboard {
        totalFame INT(15),
        rank int(10) UNIQUE,
        CID VARCHAR(20),
        AccName VARCHAR(20)
        name VARCHAR(20),
        PRIMARY KEY (CID, AccName),
```

```
        FOREIGN KEY(CID, AccName, name) REFERENCES characters(ID, AccName, name) ON DELETE
CASCADE
}

CREATE TABLE LocalLeaderboard {
        rank INT(10),
        ID VARCHAR(20),
        Fame INT(15),
        CID VARCHAR(20),
        AccName VARCHAR(20),
        location VARCHAR(20),
        PRIMARY KEY (ID, rank),
        FOREIGN KEY (CID, AccName) REFERENCES Characters(ID, AccName) ON DELETE CASCADE
        FOREIGN KEY (location) REFERENCES Server(location) ON DELETE CASCADE ON UPDATE CASCADE
}

CREATE TABLE Server {
        name VARCHAR(20),
        location VARCHAR(20),
        capacity INT(6),
        PRIMARY KEY (name, location)
}
```

University of British Columbia, Vancouver
Department of Computer Science

8. INSERT statements to populate each table with at least 5 tuples. You will likely want to have more than 5 tuples so that you can have meaningful queries later.

```
INSERT INTO Accounts
        (username, isVerified, password, email, InvSlots)
VALUES
        ('gamer', TRUE, 'password123', 'gamerboi@gmail.com', 150),
        ('lucky', FALSE, 'password123', 'luckman47@gmail.com', 150),
        ('MJJR', TRUE, 'password123', 'thekingofpop@yahoo.com', 150),
        ('cookie', FALSE, 'password123', 'cookiebanookie@gmail.com', 150),
        ('BlueMan', TRUE, 'password123', 'redguy@outlook.com', 150);

INSERT INTO Friends
        (user, friend)
VALUES
        ('gamer', 'lucky'),
        ('gamer', 'MJJR'),
        ('lucky', 'cookie'),
        ('MJJR', 'BlueMan'),
        ('cookie', 'BlueMan');

INSERT INTO Characters
        (ID, AccName, name, class, age, height, weight, race, level, money, strength, intelligence, charisma, dexterity, luck,
InvSlots)
VALUES
        ('mqsxWbYOxtZuBuaWGoac', 'gamer', 'Orbulon the Opal', 'Mage', 36, 130, 30, 'Halfling', 5, 100, 10, 12, 8, 7, 6, 50),
        ('nrANMIVY1aBiBMwVqeBJ', 'gamer', 'Caduceus Jones', 'Thief', 24, 190, 30, 'Elf', 4, 150, 8, 10, 6, 10, 7, 50),
        ('E4HhsZBMPOJLL31rTvs5', 'cookie', 'Cookie', 'Warrior', 30, 144, 105, 'Dwarf', 3, 50, 9, 7, 6, 8, 5, 50),
        ('MuWu21WHmoX4iYIVqpLb', 'BlueMan', 'Blumbo', 'Mage', 36, 178, 70, 'Human', 7, 200, 6, 12, 8, 5, 9, 50),
        ('fPsbkVEoKIZTYiKHkSqa', 'lucky', 'Ferrite Bueller', 'Pirate', 36, 137, 28, 'Halfling', 5, 120, 9, 7, 6, 8, 5, 50);

INSERT INTO Equipped
        (EqName, CID, AccName, EqType)
VALUES
        ('Sword', 'mqsxWbYOxtZuBuaWGoac', 'gamer', 'Weapon'),
        ('Ef Bow', 'nrANMIVY1aBiBMwVqeBJ', 'gamer', 'Weapon'),
        ('Ancient Axe', 'E4HhsZBMPOJLL31rTvs5', 'cookie', 'Weapon'),
        ('Magic Staff', 'MuWu21WHmoX4iYIVqpLb', 'BlueMan', 'Weapon'),
        ('Pirate Cutlass', 'fPsbkVEoKIZTYiKHkSqa', 'lucky', 'Weapon');

INSERT INTO Inventory
        (CID, AccName, slotNum, stacks, item)
VALUES
        ('mqsxWbYOxtZuBuaWGoac', 'gamer', 1, 10, 'Health Potion'),
        ('nrANMIVY1aBiBMwVqeBJ', 'gamer', 2, 5, 'Arrow'),
        ('E4HhsZBMPOJLL31rTvs5', 'cookie', 3, 2, 'Stamina Potion'),
        ('MuWu21WHmoX4iYIVqpLb', 'BlueMan', 4, 3, 'Magic Ring'),
        ('fPsbkVEoKIZTYiKHkSqa', 'lucky', 5, 1, 'Treasure Map');

INSERT INTO SharedInventory
        (userID, slotNum, stacks, item)
VALUES
        ('gamer', 1, 10, 'Rare Gem'),
        ('lucky', 2, 5, 'Mystic Herb'),
        ('MJJR', 3, 3, 'Ancient Scroll'),
        ('cookie', 4, 2, 'Healing Stone'),
        ('BlueMan', 5, 1, 'Golden Coin');
```

University of British Columbia, Vancouver
Department of Computer Science

```
INSERT INTO Items
        (name, isKey, description)
VALUES
        ('Health Potion', FALSE, 'Restores health'),
        ('Arrow', FALSE, 'Ammunition for bow'),
        ('Stamina Potion', FALSE, 'Increases stamina'),
        ('Magic Ring', TRUE, 'A ring with magical properties'),
        ('Treasure Map', TRUE, 'A map leading to hidden treasure');

INSERT INTO Consumables
        (name, maxStack)
VALUES
        ('Health Potion', 20),
        ('Stamina Potion', 10);

INSERT INTO Equipments
        (name, type, strength, intelligence, charisma, dexterity, luck)
VALUES
        ('Sword', 'Weapon', 10, 0, 0, 2, 1),
        ('Elf Bow', 'Weapon', 8, 0, 0, 10, 3),
        ('Ancient Axe, 'Weapon', 9, 0, 0, 1, 2),
        ('Magic Staff', 'Weapon', 2, 12, 0, 1, 4),
        ('Pirate Cutlass', 'Weapon', 7, 0, 0, 8, 5);

INSERT INTO Resources
        (name, maxStack)
VALUES
        ('Rare Gem', 50),
        ('Mystic Herb', 100),
        ('Ancient Scroll', 10),
        ('Healing Stone', 20),
        ('Golden Coin', 1000);

INSERT INTO GlobalLeaderboard
        (totalFame, rank, CID, AccName, name)
VALUES
        (1000, 1, 'mqsxWbYOxtZuBuaWGoac', 'gamer, 'Orbulon the Opal'),
        (900, 2, 'nrANMIVY1aBiBMwVqeBJ', 'gamer', 'Caduceus Jones'),
        (1100, 3, 'MuWu21WHmoX4iYIVqpLb', 'BlueMan', 'Blumbo'),
        (800, 4, 'E4HhsZBMPOJLL31rTvs5', 'cookie', 'Cookie'),
        (950, 5, 'fPsbkVEoKIZTYiKHkSqa', 'lucky', 'Ferrite Bueller');

INSERT INTO LocalLeaderboard
        (rank, ID, Fame, CID, AccName, location)
VALUES
        (1, 'loc1', 500, 'mqsxWbYOxtZuBuaWGoac', 'MJJR', 'North America'),
        (2, 'loc2', 450, 'nrANMIVY1aBiBMwVqeBJ', 'gamer', 'South America'),
        (3, 'loc3', 550, 'MuWu21WHmoX4iYIVqpLb', 'BlueMan', 'Africa'),
        (4, 'loc4', 400, 'E4HhsZBMPOJLL31rTvs5', 'cookie', 'Asia'),
        (5, 'loc5', 420, 'fPsbkVEoKIZTYiKHkSqa', 'lucky', 'Europe');

INSERT INTO Server
        (name, location, capacity)
VALUES
        ('Server1', 'Asia', 1000),
        ('Server2', 'Europe', 800),
        ('Server3', 'North America, 1200),
        ('Server4', 'South America', 600),
        ('Server5', 'Africa, 700);
```